IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re application of: Richard Alan Barraclough

Filed: Dec. 27, 2001

Serial Number: 10/034467

Title: Wizard Builder For Application Software

Examiner: Ryan F. Pitaro

Group Art Unit: 2174

Declaration Under Rule 131

United Kingdom

County of Cheshire

I, Richard Alan Barraclough, do hereby declare and say the following.

My address is 28 Lacey Court, Wilmslow, Cheshire SK9 4BH, United Kingdom

I have am the Inventor of the Invention in the above Application, which is assigned to ExperTune, Inc. Attached to this Declaration is a true copy of a document prepared by me and labeled Exhibit A showing that the Invention of the above Application was completed by me, by both conception and reduction to practice, by building and testing, in the United Kingdom at least as early as November 15, 2000. Exhibit A also shows how parts of the program and screen shots relate to Claim 1 of the Application.

I further declare that all statements made herein of my own knowledge are true, and that all statements made on information and belief, are believed to be true; and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the patent application or any patents issuing therefrom.

R A Barrac

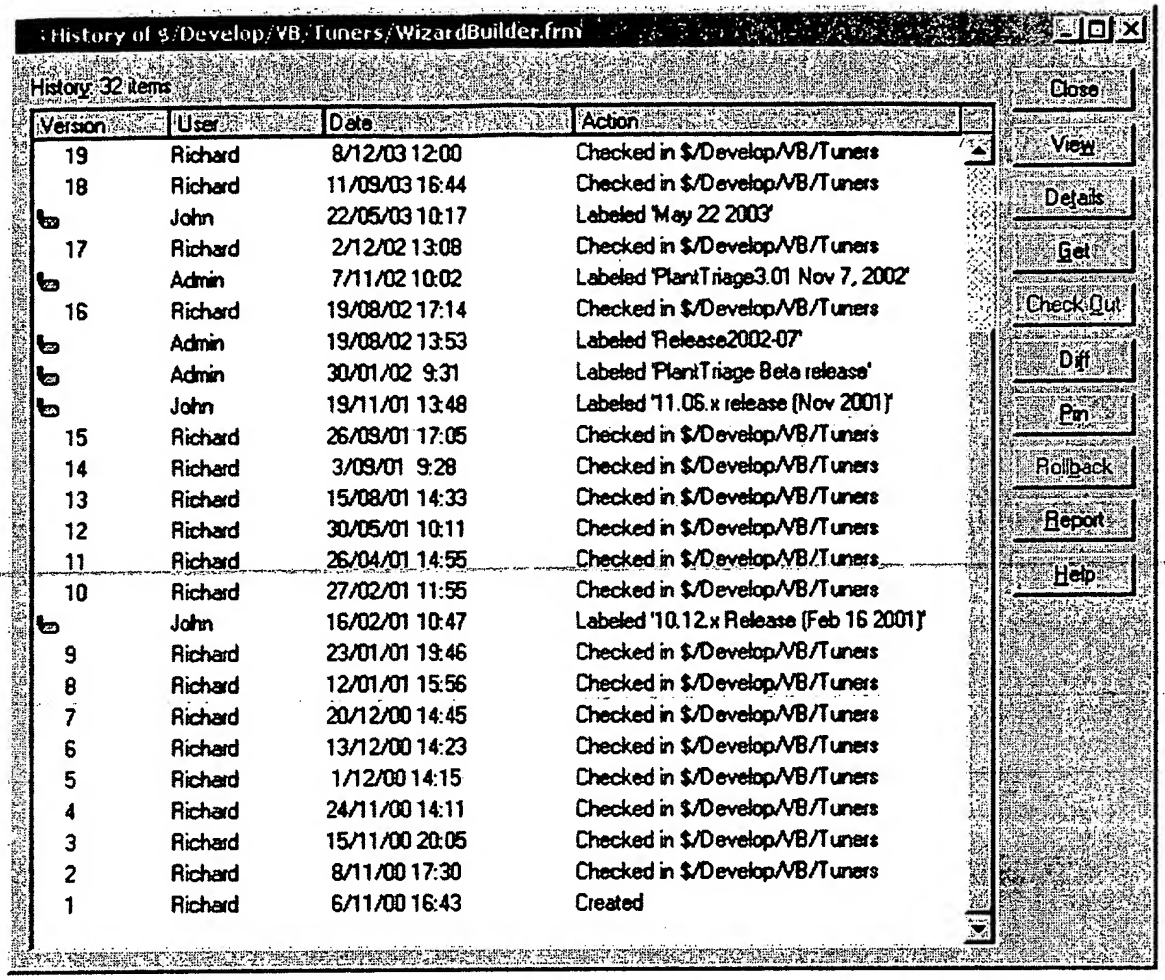
Richard Alan Barraclough, Inventor

6 October 2005

(date)

EXHIBIT A

Figure 1. Date of initial creation of the wizard building invention.



Version	User	Date	Action
19	Richard	8/12/03 12:00	Checked in $/Develop/VB/Tuners
18	Richard	11/09/03 16:44	Checked in $/Develop/VB/Tuners
	John	22/05/03 10:17	Labeled 'May 22 2003'
17	Richard	2/12/02 13:08	Checked in $/Develop/VB/Tuners
	Admin	7/11/02 10:02	Labeled 'PlantTriage3.01 Nov 7, 2002'
16	Richard	19/08/02 17:14	Checked in $/Develop/VB/Tuners
	Admin	19/08/02 13:53	Labeled 'Release2002-07'
	Admin	30/01/02 9:31	Labeled 'PlantTriage Beta release'
	John	19/11/01 13:48	Labeled '11.06.x release (Nov 2001)'
15	Richard	26/09/01 17:05	Checked in $/Develop/VB/Tuners
14	Richard	3/09/01 9:28	Checked in $/Develop/VB/Tuners
13	Richard	15/08/01 14:33	Checked in $/Develop/VB/Tuners
12	Richard	30/05/01 10:11	Checked in $/Develop/VB/Tuners
11	Richard	26/04/01 14:55	Checked in $/Develop/VB/Tuners
10	Richard	27/02/01 11:55	Checked in $/Develop/VB/Tuners
	John	16/02/01 10:47	Labeled '10.12.x Release (Feb 16 2001)'
9	Richard	23/01/01 19:46	Checked in $/Develop/VB/Tuners
8	Richard	12/01/01 15:56	Checked in $/Develop/VB/Tuners
7	Richard	20/12/00 14:45	Checked in $/Develop/VB/Tuners
6	Richard	13/12/00 14:23	Checked in $/Develop/VB/Tuners
5	Richard	1/12/00 14:15	Checked in $/Develop/VB/Tuners
4	Richard	24/11/00 14:11	Checked in $/Develop/VB/Tuners
3	Richard	15/11/00 20:05	Checked in $/Develop/VB/Tuners
2	Richard	8/11/00 17:30	Checked in $/Develop/VB/Tuners
1	Richard	6/11/00 16:43	Created

Figure 1 is a screen copy of a Microsoft utility named SourceSafe that is used by ExperTune Inc. SourceSafe is designed for software development and is widely used within the software development industry. It keeps secure all versions of all source files. Figure 1 shows that the file WizardBuilder.frm was first created 6 November 2000. Note that the dates are in UK format (day/month/year). WizardBuilder.frm is the source code for the wizard builder invention. Compiling this source code creates an executable file which can run and respond to a human user. Figure 1 also shows that the wizard builder invention was continually altered and refined. It was first released to customers 16 February 2001. A useful feature of SourceSafe is that any version of any file can be recovered.

Figure 2. Date of initial creation of help screen for wizard building invention.

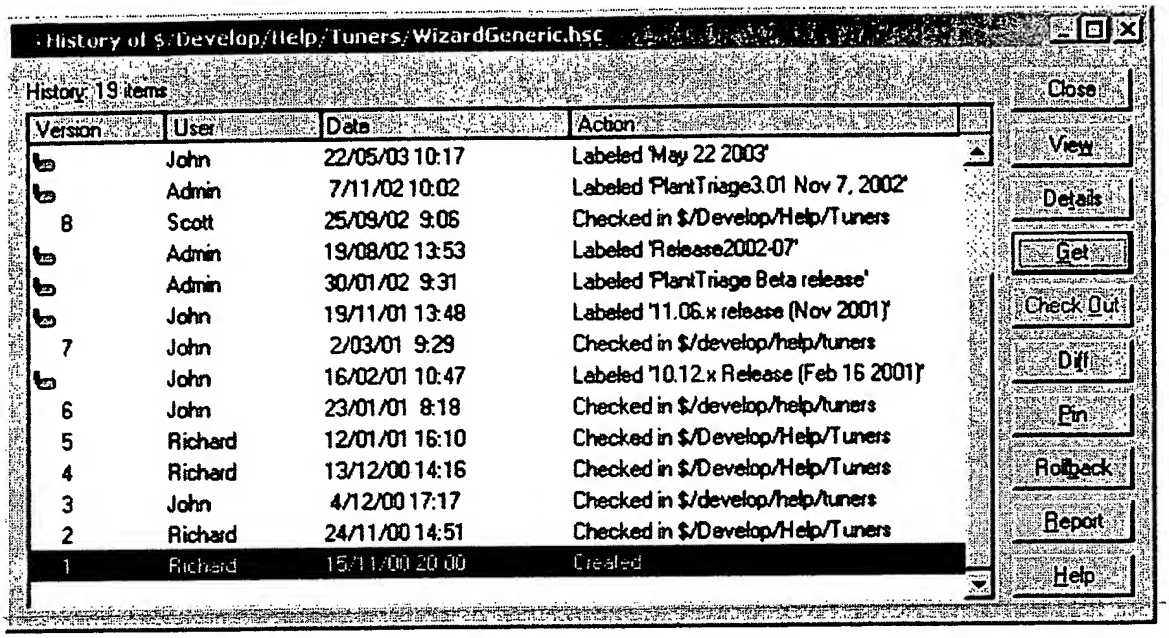


Figure 2 is another SourceSafe screen in use by ExperTune Inc. This screen shows the history of the WizardGeneric.hsc file. This file is the source code for the user help screens for the wizard builder invention. Note that this file was first created 15 November 2000.

Figure 3. User Help screen for wizard building invention.

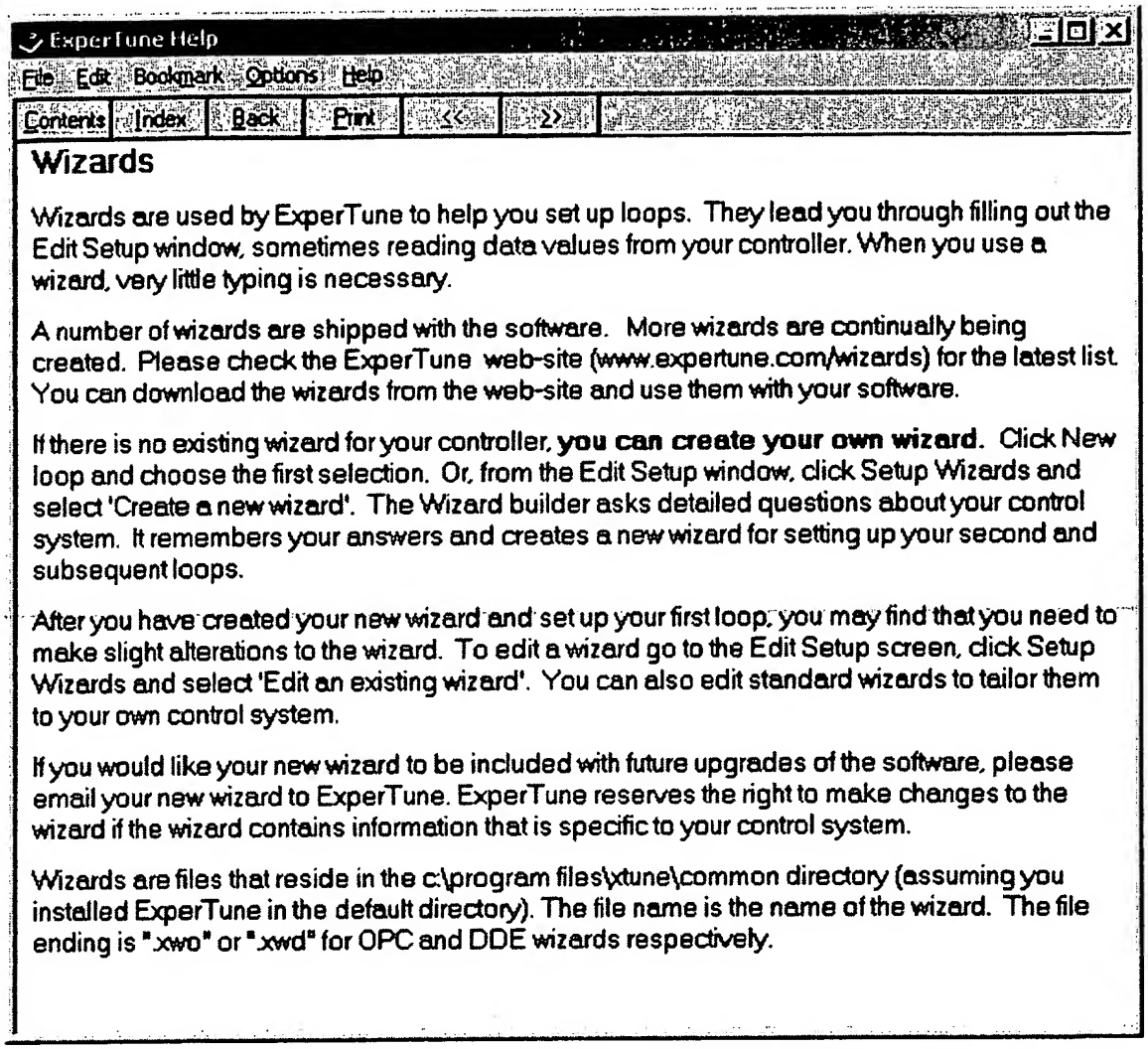


Figure 3 is a user help screen for one embodiment of the wizard builder invention. The embodiment is for setting up a DDE (Dynamic Data Exchange) de-facto interface whereby application software is set up to communicate with industrial control and monitoring equipment (loop controllers).

This help screen has been derived from the actual WizardGeneric.hsc source file that was created 15 November 2000. The version dated 15 November 2000 was recovered from the ExperTune Inc. SourceSafe system and re-compiled. Note the last two sentences of the third paragraph which have been proved to have been written prior to 15 November 2000: "The Wizard builder asks detailed questions about your (A)control system. It remembers your answers and (B) creates a new wizard for setting up your second and subsequent loops". These two sentences summarize the wizard builder invention in terms meaningful to a human user of the invention.

Compare this description with claim 1: "A computer program process, called a wizard builder, executable on a computer, for adapting an application program to function with (1)devices or sensors monitoring or controlling a process occurring in real-time, wherein the wizard builder (2) constructs a setup wizard which sets up (4) a de facto interface between the devices or sensors and the application program, wherein (3) the setup wizard is constructed by means of asking a

human user of the application program simple verbal questions, in English or other language convenient to a human user, about the model and manufacturer of the devices or sensors, and wherein the wizard builder does not require a human user to have any knowledge of writing or using computer programs."

The user help screen uses the term "(A)control system", instead of the more lengthy "(1)devices or sensors monitoring or controlling a process occurring in real-time". The user help screen then states "(B) creates a new wizard for setting up your second and subsequent loops". This corresponds to the claim 1 statement, "(2) constructs a setup wizard which sets up a de facto interface between the devices or sensors and the application program"

To further show that the wizard builder invention existed in November 2000, we have taken the source file, WizardBuilder.frm, that was current 24 November 2000. This is version 4 of the source file. See figure 1. The source file from this date was recovered from SourceSafe, compiled and run. Figures 4 to 32 below clearly show "(3) the setup wizard is constructed by means of asking a human user of the application program simple verbal questions, in English or other language convenient to a human user, about the model and manufacturer of the devices or sensors, and wherein the wizard builder does not require a human user to have any knowledge of writing or using computer programs". The following screen shots have therefore been proved to exist prior to 24 November 2000.

Figure 4. User chooses to to create a new wizard.

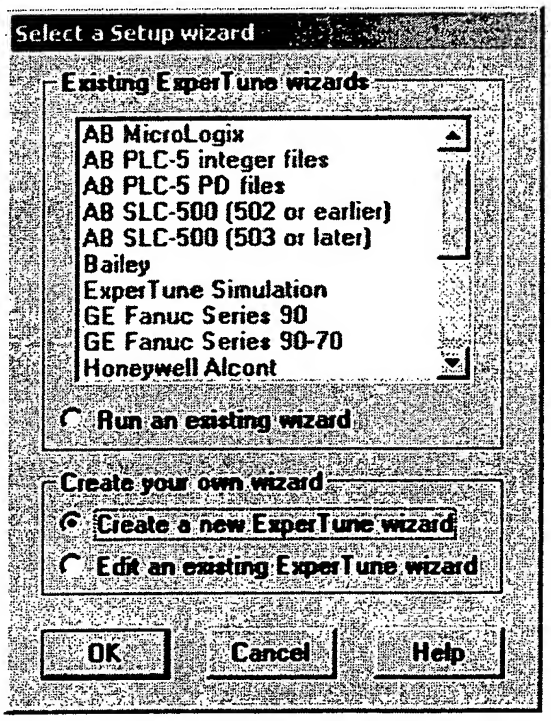


Figure 4 is the first screen shown to the user when he wishes to use the wizard builder invention. He has the choice of using the wizard builder invention to create a new setup wizard, to run a setup wizard that has already been created, or to edit an existing setup wizard. In this case he chooses to create a new setup wizard.

Figure 5. User enters a name for the new setup wizard.

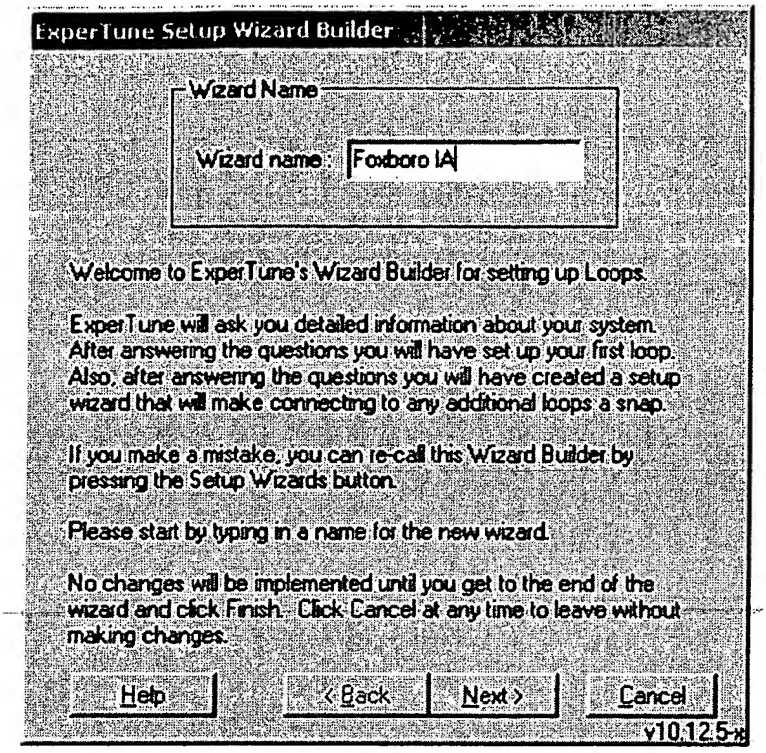


Figure 5 is the screen that first appears when the user has elected to use the wizard builder invention to create a new setup wizard. He is asked in plain English for the name of the new wizard. In this case he chooses the name Foxboro IA.

Figure 6. Wizard Builder – DDE Application Name

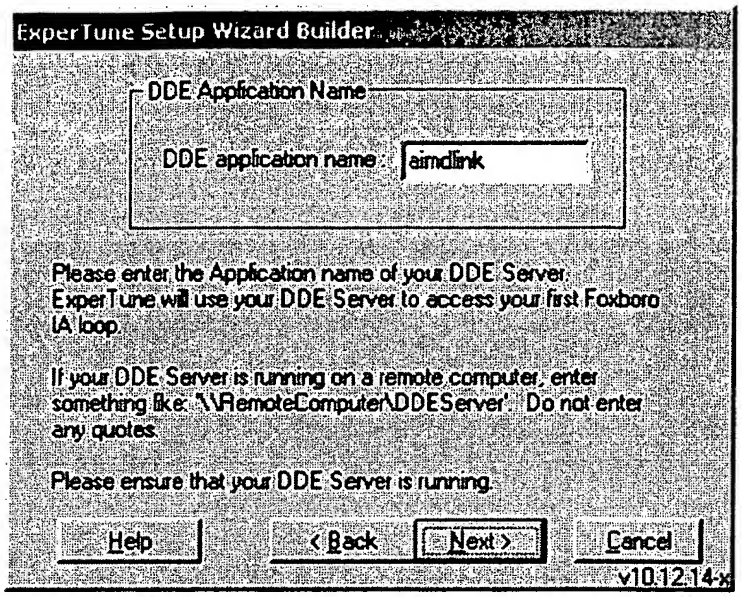


Figure 6 is where the wizard builder invention asks the user for the name of a DDE server. DDE is a well known to process and instrument engineers as a "(4) de facto interface between the devices or sensors and the application program". In fact DDE is defined and supported by Microsoft. In this instance the user has chosen the name "aimdlink". The user does not need knowledge of writing or using computer programs to come up with this name. The DDE server user documentation will clearly show the DDE Application name which the DDE server uses.

Figure 7. Wizard Builder – Fixed DDE Application?

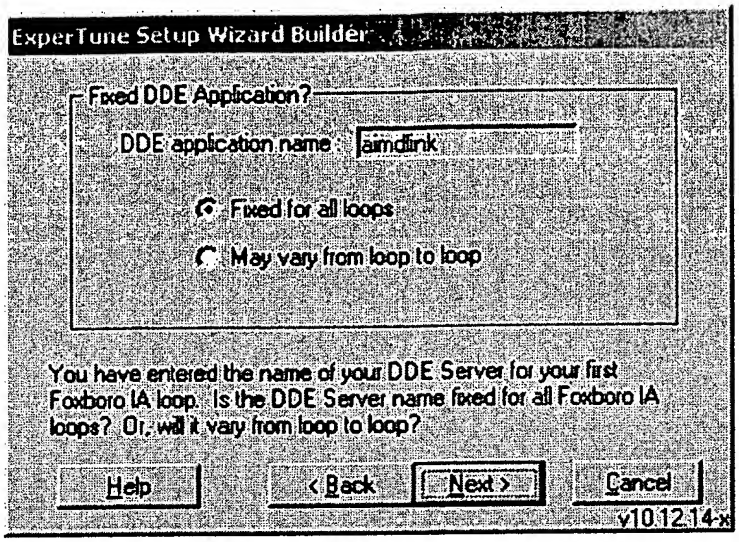


Figure 7 is where the wizard builder invention asks the user for further information about the DDE interface. The user has the choice of having the resultant setup wizard ask for a new DDE Application Name or to remember the DDE Application name everytime the resultant setup wizard runs. In this instance the user indicates that the setup wizard should not ask for a new DDE application name but should always use the DDE Application named "aimdlink" every time it runs.

Figure 8. Wizard Builder – DDE Topic name

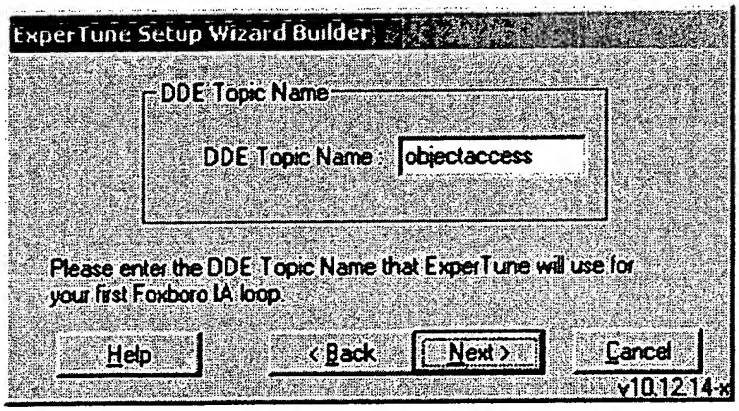


Figure 8 is where wizard builder invention asks the user for the DDE topic name. The DDE Topic name is part of the DDE de facto interface. It further refines the DDE Application name. It is a name that needs to be recognised by the DDE server. In this instance the user has chosen the name "objectaccess". The user does not need knowledge of writing or using computer programs to come up with this name. The DDE server user documentation will clearly show the DDE Topic names which the DDE server recognises.

Figure 9. Wizard Builder Invention – Fixed DDE Topic Name?

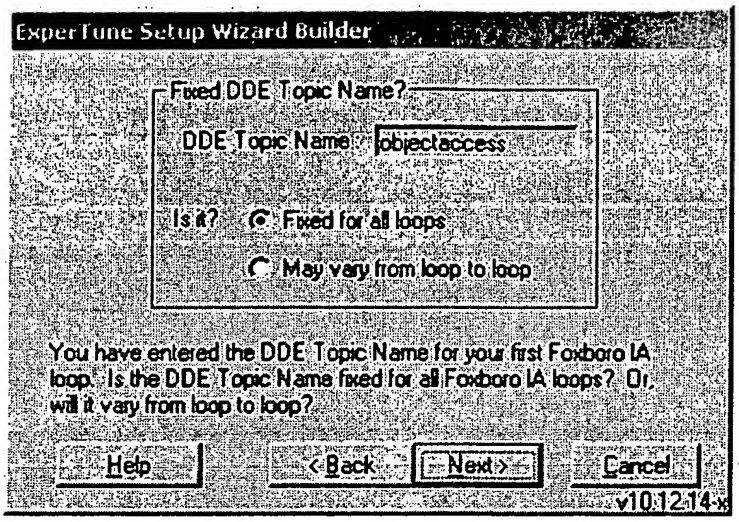


Figure 9 is where the wizard builder invention asks the user for further information about the DDE interface. The user has the choice of having the resultant setup wizard ask for a new DDE Topic Name or to remember the DDE Topic name everytime the resultant setup wizard runs. In this instance the user indicates that the setup wizard should not ask for a new DDE Topic name but should always use the DDE Topic named "objectaccess" every time it runs.

Figure 10. Wizard Builder Invention – definition of loop process variable

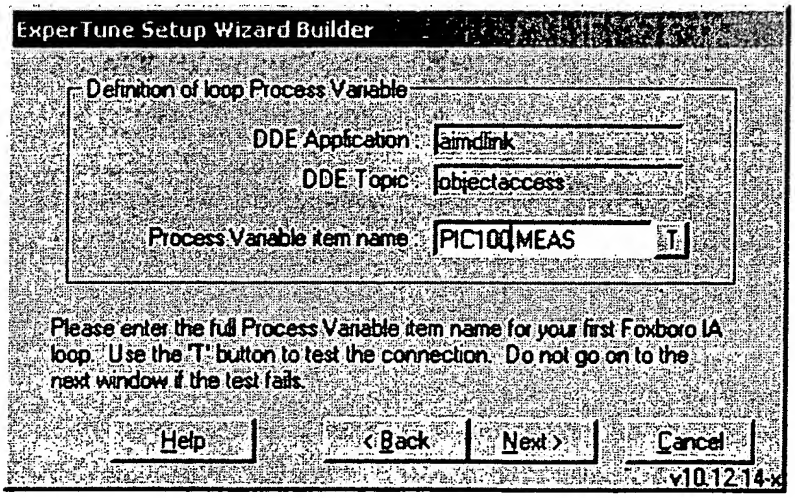


Figure 10 is where wizard builder invention asks for a DDE item name for the loop controller's process variable. In this instance the user has entered the item name: PIC100.MEAS. The user does not need knowledge of writing or using computer programs to come up with this name. The DDE server user documentation will clearly show the DDE item names which the DDE server recognises.

Figure 11. Wizard Builder Invention – Prefix prompt

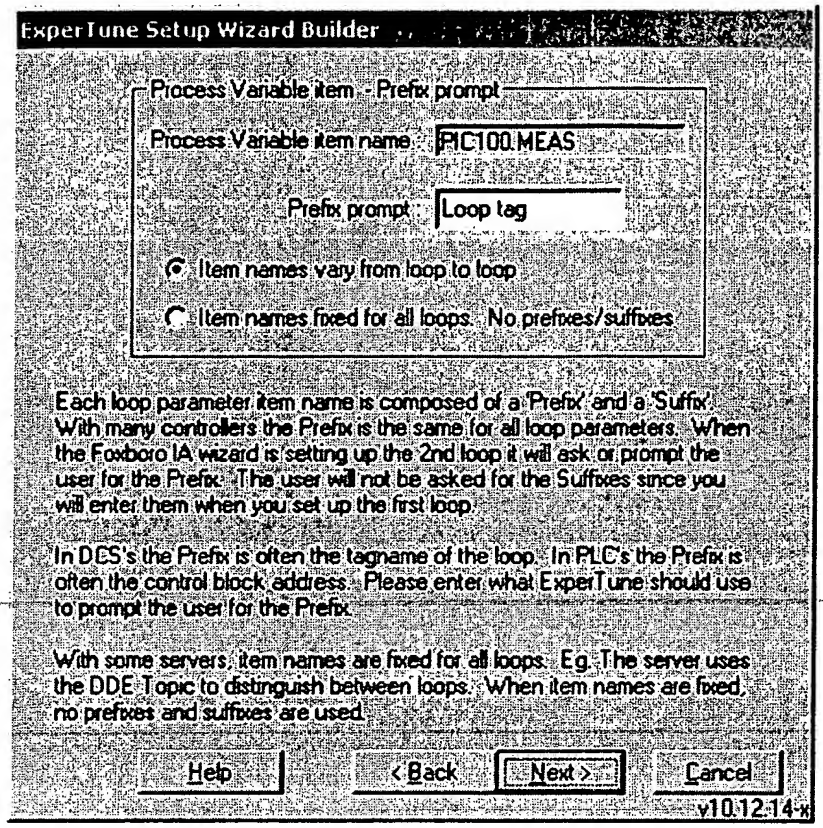


Figure 11 is where the wizard builder invention asks the user for a prefix prompt. The prefix prompt will eventually be used by the setup wizard that is created by the wizard builder invention.

Figure 12. Wizard Builder Invention – Parameter suffixes

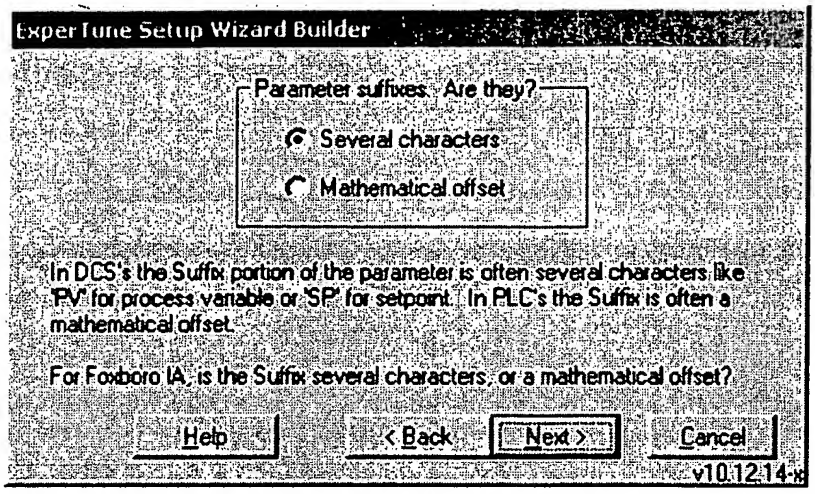


Figure 12 is where the wizard builder invention asks the user for the type of suffixes used. The user does not need knowledge of writing or using computer programs to answer this question. The DDE server user documentation will clearly show the DDE item name structure and hence the type of suffix which the setup wizard, that is being created by the wizard builder invention, should use.

Figure 13. Wizard Builder Invention – Prefix/suffix split

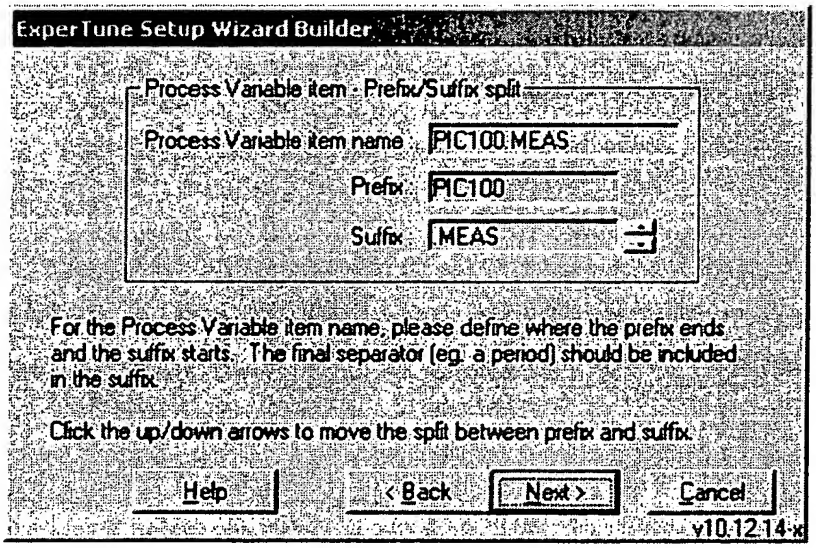


Figure 13 is where the wizard builder invention asks the user where in the process variable DDE item name the prefix/suffix split should occur. This information will be used by the setup wizard, that is being created by the wizard builder invention. When the setup wizard runs it will ask the user for a new prefix value and automatically add the suffix to create a new process variable DDE item name for another loop controller. The user does not need knowledge of writing or using computer programs to answer this question. The DDE server user documentation will clearly show the DDE item name structure and hence the suffixes that are common to all loop controllers.

Figure 14. Wizard Builder Invention – Suffix for Controller Output DDE item names.

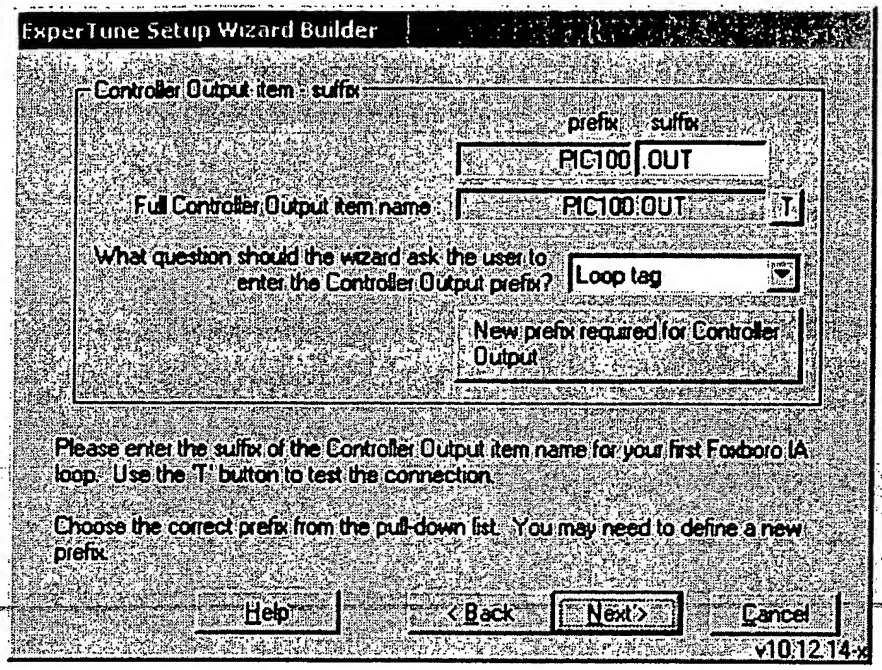


Figure 15. Wizard Builder Invention – Suffix for Setpoint DDE item names.

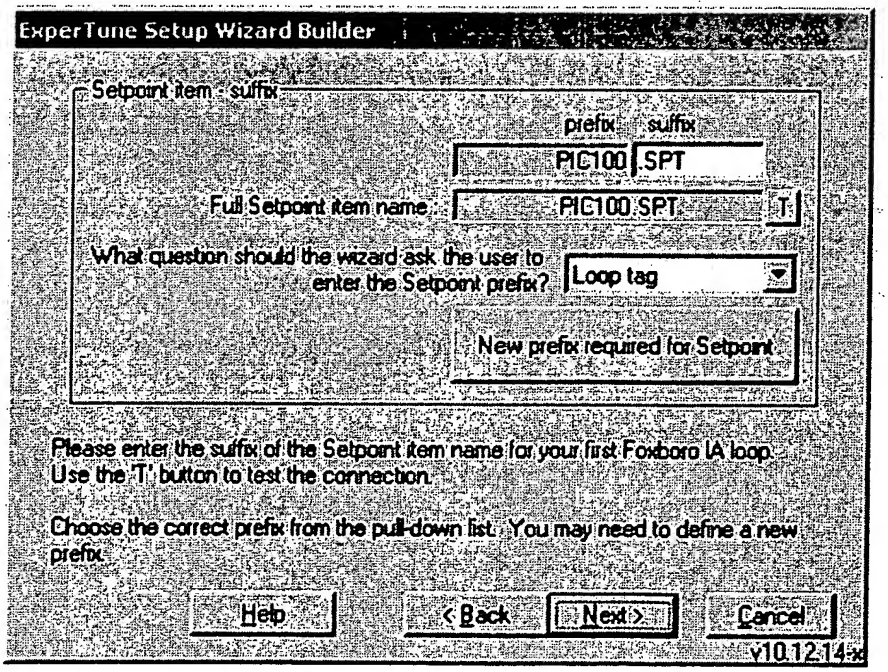


Figure 16. Wizard Builder Invention – Suffix for P term DDE item names.

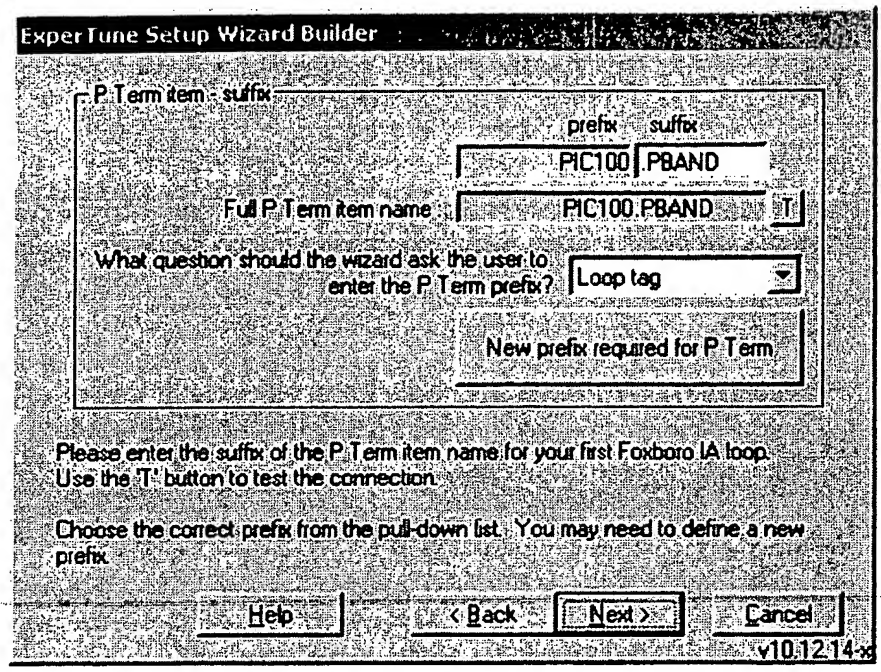


Figure 17. Wizard Builder Invention – Suffix for I term DDE item names.

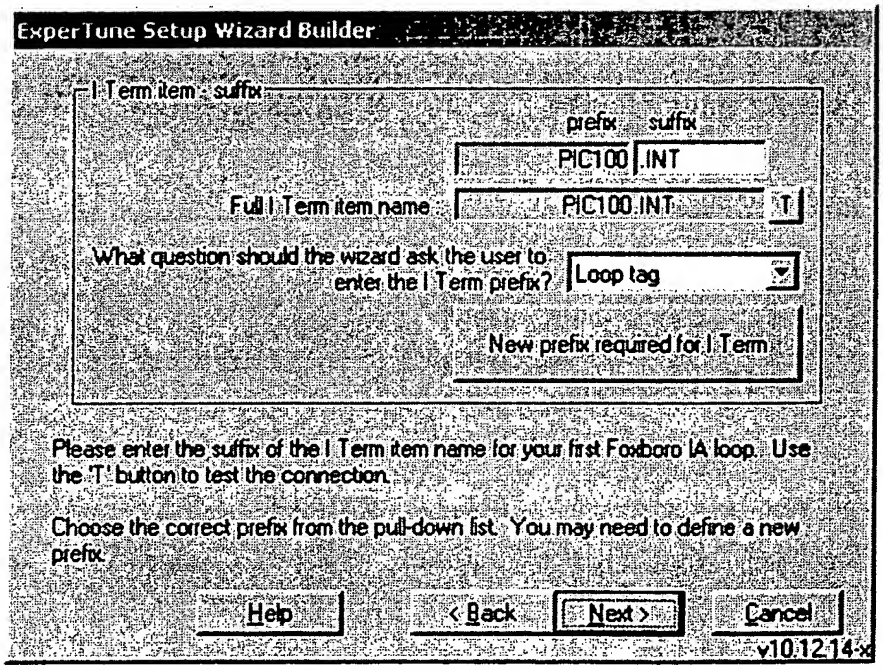


Figure 18. Wizard Builder Invention – Suffix for D term DDE item names.

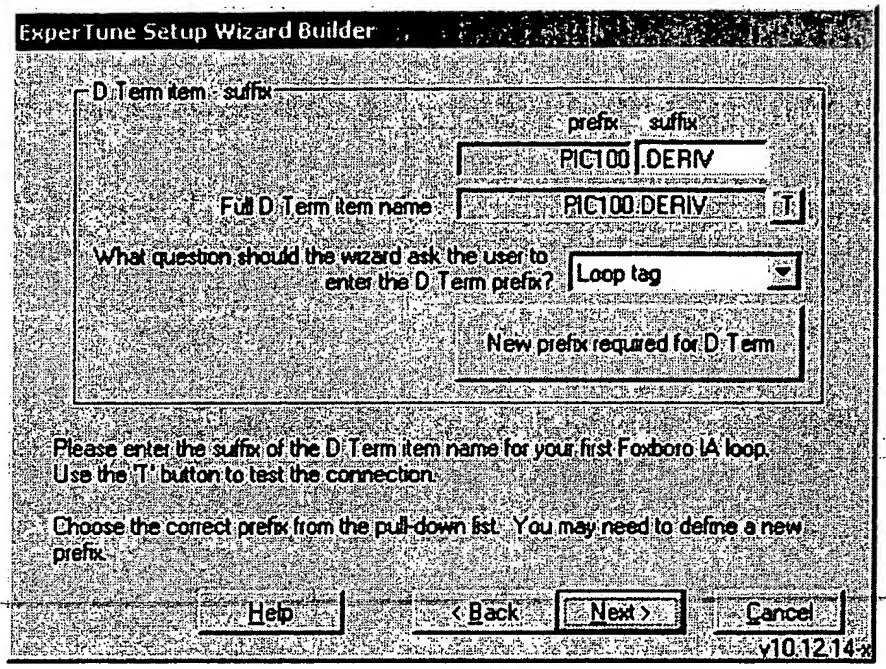
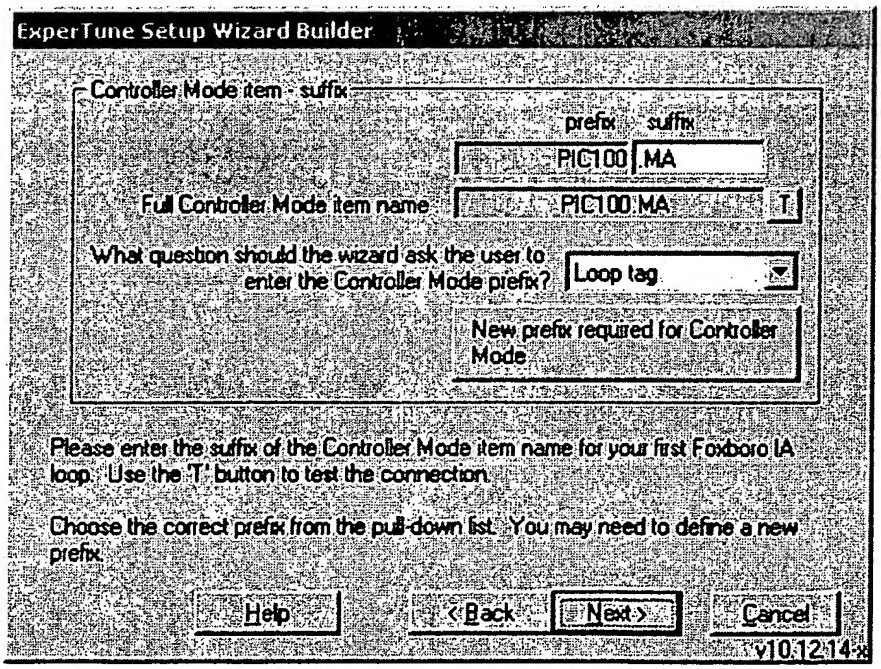


Figure 19. Wizard Builder Invention – Suffix for Controller mode DDE item names.



Figures 14 to 19 is where the wizard builder invention asks the user for the suffixes for other loop controller DDE item names. This information will be used by the setup wizard, that is being created by the wizard builder invention. When the setup wizard runs it will ask the user for a new prefix value and automatically add the different suffixes for the different loop controller parameters. The user does not need knowledge of writing or using computer programs to answer

this question. The DDE server user documentation will clearly show the DDE item name structure and hence the suffixes that are common to all loop controllers.

Figure 20. Wizard Builder Invention – Auto and Manual modes.

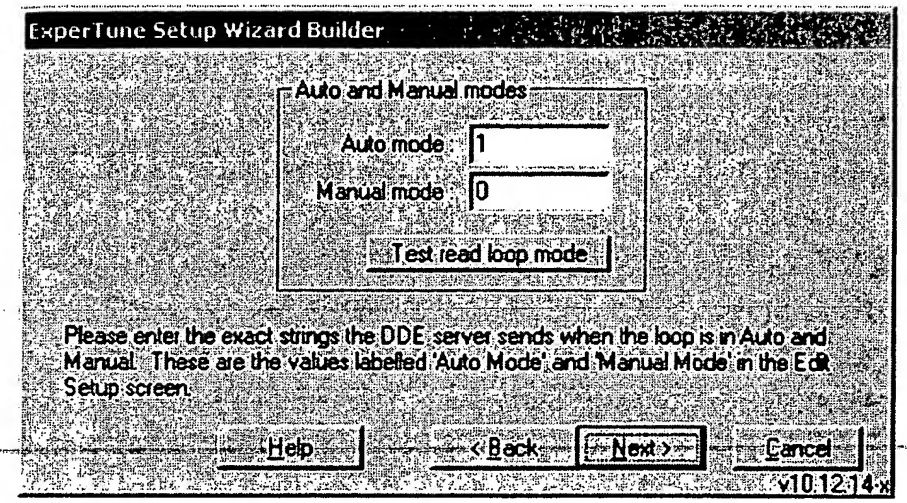


Figure 20 is where the wizard builder invention asks the user for the exact strings the DDE server sends for different controller loop modes. In this case, the user has entered "1" for auto mode and "0" for manual mode. This information will be used by the setup wizard, that is being created by the wizard builder invention. When the setup wizard runs it will automatically set up these exact strings. The user does not need knowledge of writing or using computer programs to answer this question. He can simply use the "Test read loop mode" button to manually examine the strings being sent by the DDE server.

Figure 21. Wizard Builder Invention – Unique write locations

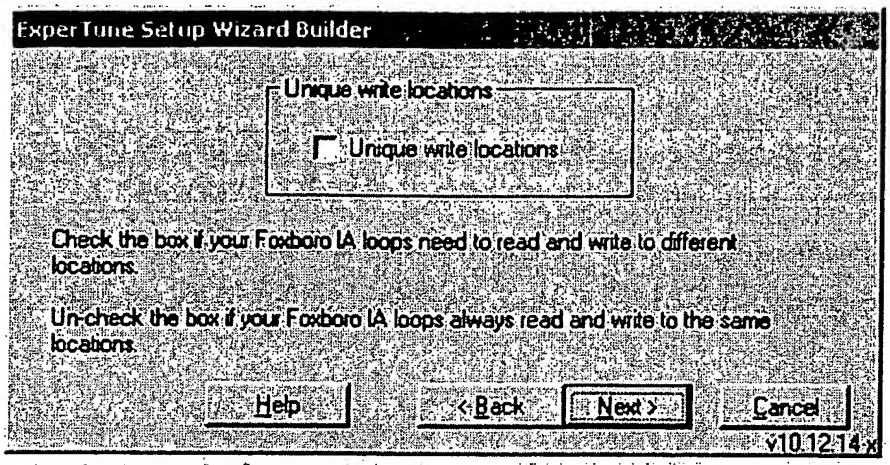


Figure 21 is where the wizard builder invention asks the user if his loop controller uses two different DDE item names: one for reading and one for writing. In this instance the user has indicated that the loop controller uses the same address for both reading and writing. The user does not need knowledge of writing or using computer programs to answer this question. The DDE server user documentation will clearly show the DDE item names to be used for read and write access to the loop controller.

Figure 22. Wizard Builder Invention – PV and CO scaling

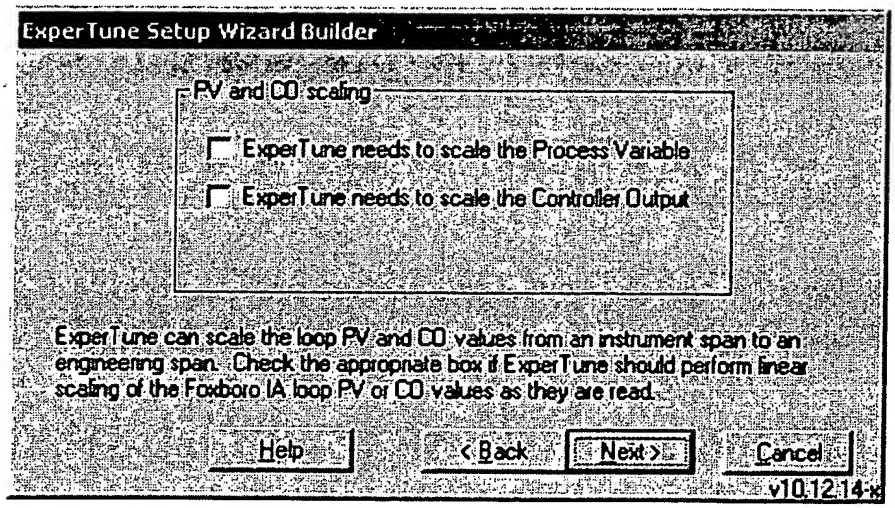


Figure 22 is where the wizard builder invention asks the user if the Process Variable or Controller Output values being input from the loop controller need to be scaled. In this instance neither values need scaling. The user does not need knowledge of writing or using computer programs to answer this question. He will have already tested the process variable and controller output values in figures 10 and 14 and determined if the values are being shown correctly. Making different choices in figure 22 causes the wizard builder invention to ask different subsequent questions.

Figure 23. Wizard Builder Invention – PV engineering span

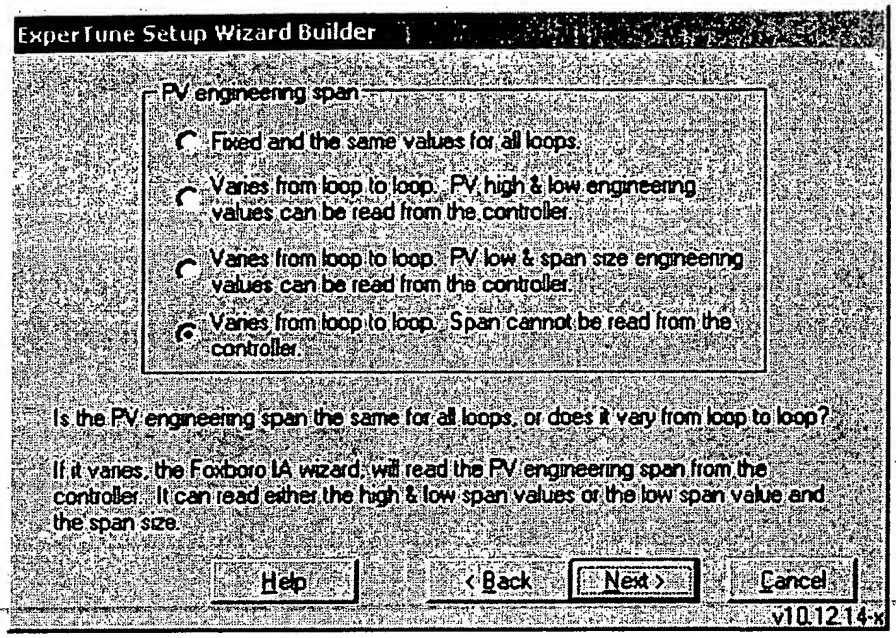


Figure 23 is where the wizard builder invention asks the user how the Process Variable engineering span should be set up. It can be either fixed, read from the loop controller or manually entered. Whatever the answer, the resultant set up wizard, being created by the wizard builder invention will, act accordingly. In this instance the Process Variable span needs to be entered manually. Therefore the resultant setup wizard will ask the user for the span values every time it runs. The user does not need knowledge of writing or using computer programs to answer this question. The DDE server user documentation will clearly show the DDE item names available for each loop controller and in this instance the span values are not available from the DDE server.

Figure 24. Wizard Builder Invention – PV engineering span values

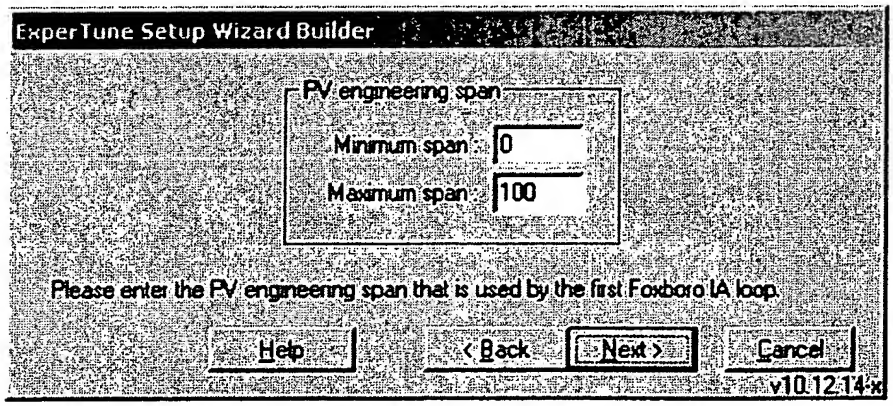


Figure 24 is where the wizard builder invention asks the user for the Process Variable engineering span values. These will be the default values presented when the Setup wizard, being created by the wizard builder invention, runs.

Figure 25. Wizard Builder Invention – CO engineering span

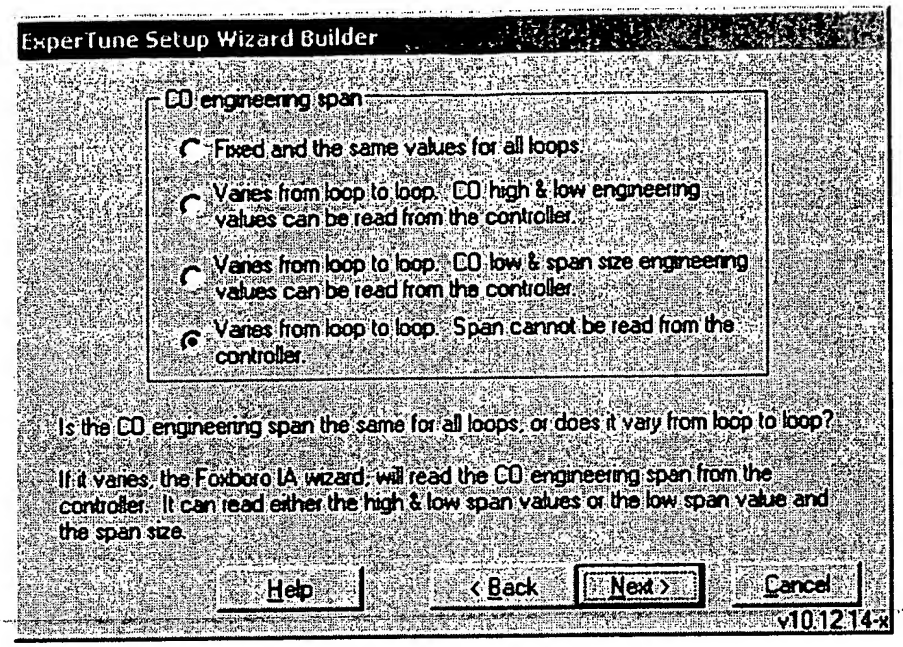
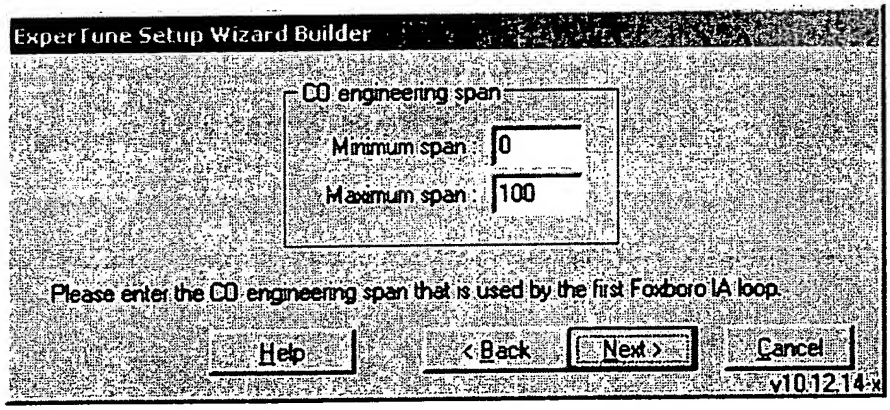


Figure 26. Wizard Builder Invention – CO engineering span values



Figures 25 and 26 are equivalent to figures 23 and 24 except the wizard builder invention is asking, in plain English, for the Controller Output span values of the loop controller instead of the Process Variable span values. The user does not need knowledge of writing or using computer programs to answer these questions. The engineering span values will be readily available from within his loop controller system.

Figure 27. Wizard Builder Invention – How to read the Controller Type

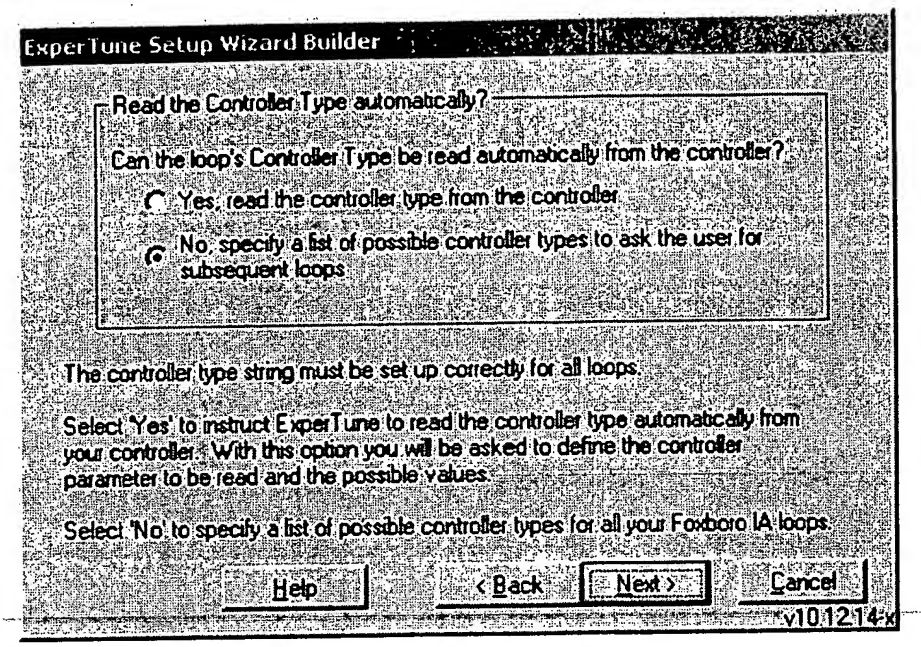


Figure 27 is where the wizard builder invention is asking, in plain English, if the resultant setup wizard, being created by the wizard builder invention, can automatically read the controller type value from the loop controller. In this instance the user has answered No. Subsequently, the wizard builder invention then allows the user to select which controller types could be valid for subsequent loop controllers. The user does not need knowledge of writing or using computer programs to answer this question. The possible controller types will be available from the loop controller documentation.

Figure 28. Wizard Builder Invention – Controller Type options

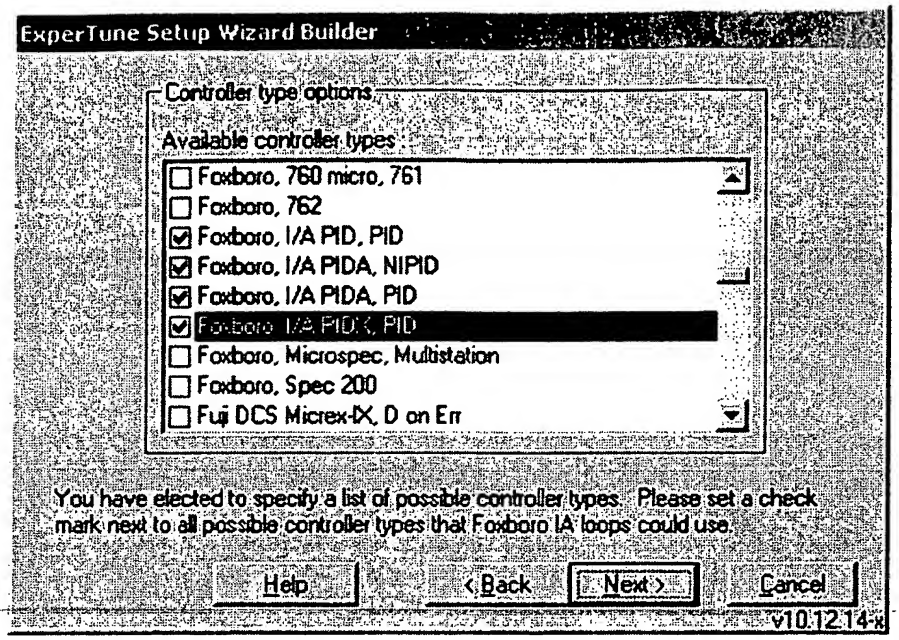


Figure 28 is where the wizard builder invention is asking the user to select the possible controller types that could be valid for the loop controller. When the resultant setup wizard, being created by the wizard builder invention, runs, it will ask the user to select from this pre-defined sub-list of controller types. The user does not need knowledge of writing or using computer programs to answer this question. The possible controller types will be available from the the loop controller documentation.

Figure 29. Wizard Builder Invention – Controller Type currently in use

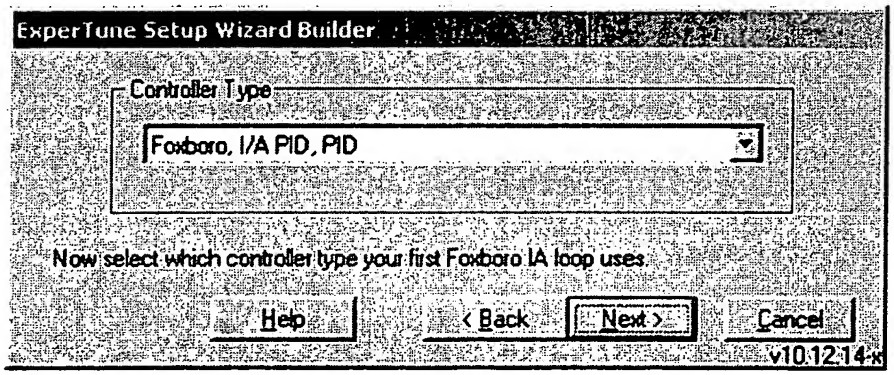


Figure 29 is where the wizard builder invention is asking, in plain English, the user to select the actual controller type in use by the current loop controller. This will be the default Controller Type presented to the user when the Setup wizard, being created by the wizard builder invention, runs.

Figure 30. Wizard Builder Invention – How to read the Process Direction

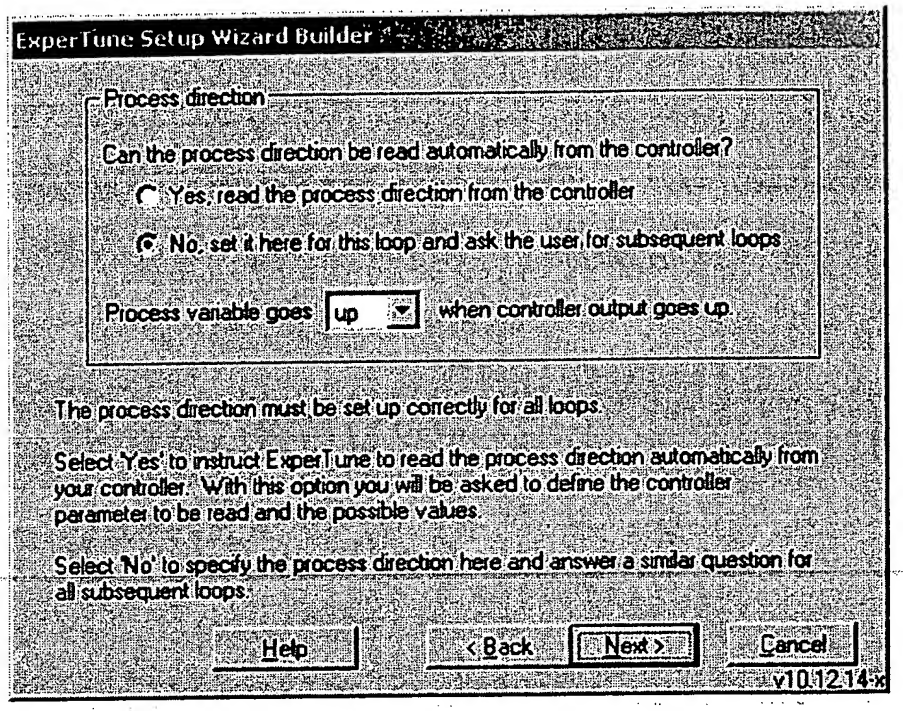


Figure 30 is where the wizard builder invention is asking, in plain English, if the resultant setup wizard, being created by the wizard builder invention, can automatically read the Process Direction value from the loop controller. In this instance the user has answered No. After the user answers No, the wizard builder invention also asks the user for the Process Direction used by the current loop controller. In this instance the user has answered "up". The user does not need knowledge of writing or using computer programs to answer this question. The Process Direction is dependent on how his process plant has been designed and built. When the resultant setup wizard, being created by the wizard builder invention, runs, it will not read the process direction from the loop controller automatically, but will ask the user to select the Process Direction. The setup wizard will default the Process Direction to "up".

Figure 31. Wizard Builder Invention – Sample Interval

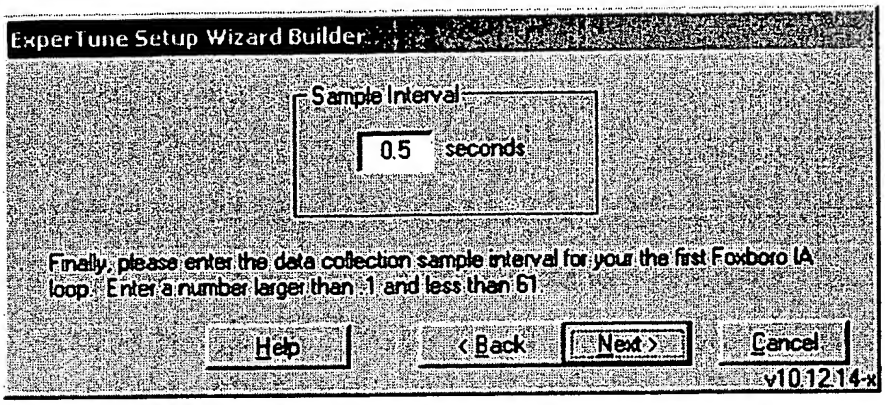


Figure 31 is where the wizard builder invention is asking, in plain English, the sample interval to be used. The user does not need knowledge of writing or using computer programs to answer this question. He does, however, need detailed knowledge of his process plant being controlled. When the resultant setup wizard, being created by the wizard builder invention, runs, it will, in this instance use "0.5" as the default sample interval.

Figure 32. Wizard Builder Invention – Summary window

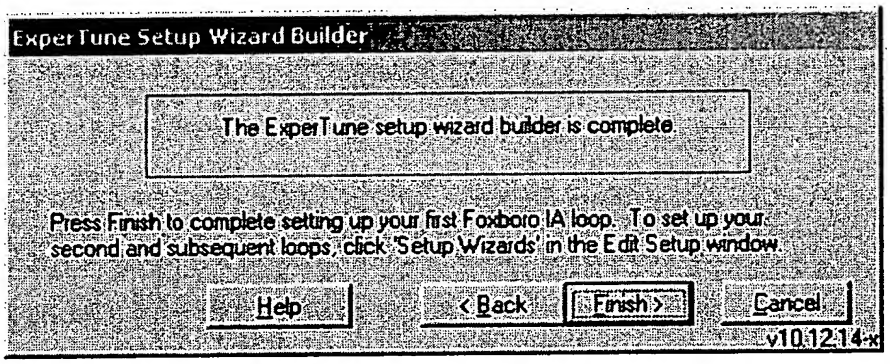


Figure 32 is the final summary window produced by the wizard builder invention. When the user clicks "Finish", the wizard builder invention will create a new setup wizard which can be subsequently run by the same user. In this instance the new setup wizard will be named "Foxboro IA".

Figure 33. User chooses to run a newly created wizard

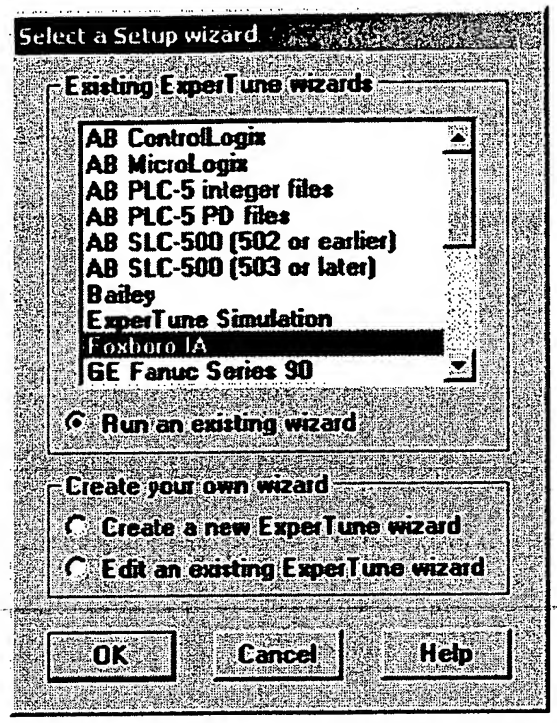


Figure 33 is a repeat of figure 4. With figure 4, the user chose to use the wizard builder invention to create a new wizard. Figures 5 to 32 are screens from the wizard builder invention where the user was using the wizard builder invention to create a new wizard named "Foxboro IA". Now that the new wizard has been created, the user can choose to run the new "Foxboro IA" wizard. Figures 34 to 41 are screenshots from running the new setup wizard that has been created by the wizard builder invention. Note that both the wizard builder and the setup wizard, that the wizard builder has created, are both run from within the same application program.

Figure 34. The created wizard asks for the loop tag.

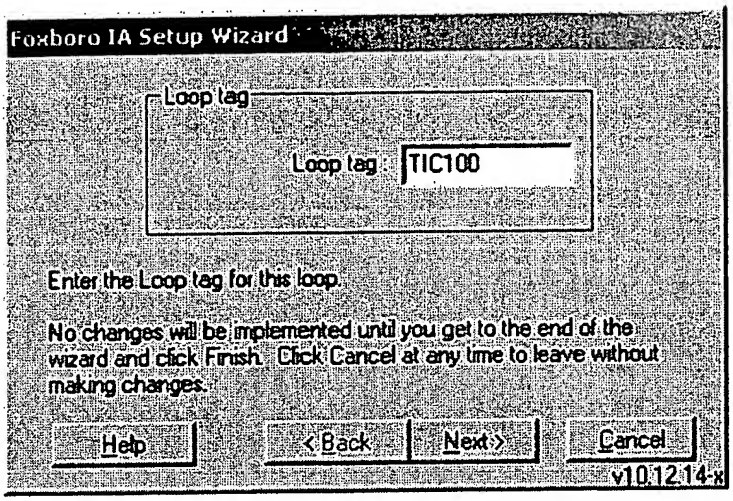


Figure 34 is where the Foxboro IA setup wizard which has been created by the wizard builder invention is asking for the loop tag name. The prompt "loop tag" was entered by the user into the wizard builder invention when the setup wizard was being created, see figure 11.

Figure 35. The created wizard asks for the PV engineering span

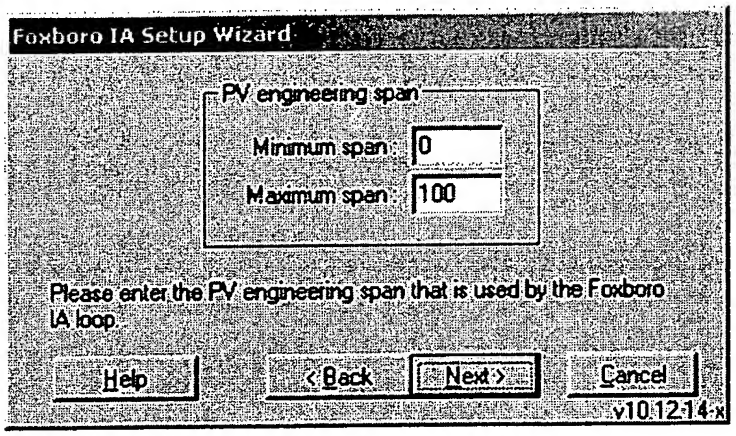


Figure 35 is where the Foxboro IA setup wizard which has been created by the wizard builder invention is asking for the Process Variable engineering span. When the user was using the wizard builder invention to create the Foxboro IA setup wizard he indicated that the PV Engineering Span could not be read automatically from the loop controller and must be entered manually. See figure 23.

Figure 36. The created wizard asks for the CO engineering span

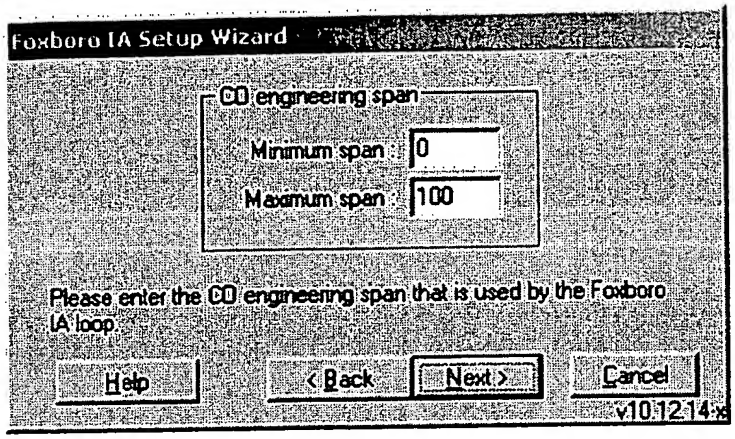


Figure 36 is where the Foxboro IA setup wizard which has been created by the wizard builder invention is asking for the Controller Output engineering span. When the user was using the wizard builder invention to create the Foxboro IA setup wizard he indicated that the CO Engineering Span could not be read automatically from the loop controller and must be entered manually. See figure 25.

Figure 37. The created wizard asks for the Controller Type

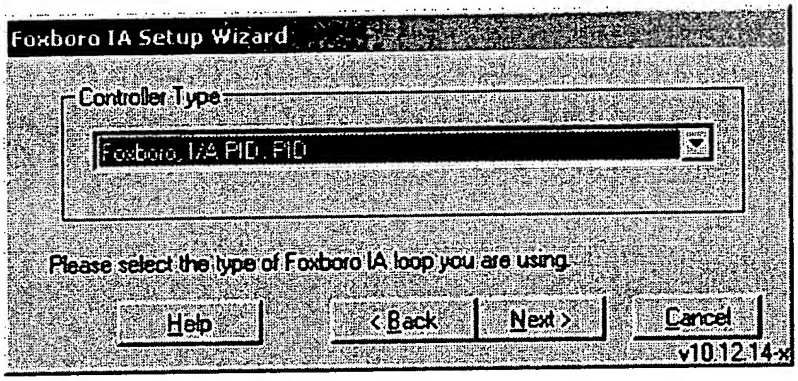


Figure 37 is where the Foxboro IA setup wizard which has been created by the wizard builder invention is asking for the Controller Type. When the user was using the wizard builder invention to create the Foxboro IA setup wizard he indicated that the Controller Type could not be read automatically from the loop controller and must be entered manually. He then indicated which controller type values were possible. See figures 27 and 28. Therefore when the Foxboro IA setup wizard is running, the controller type options available are those selected on the earlier wizard builder invention screenshot, figure 28.

Figure 38. The created wizard asks for the Process direction

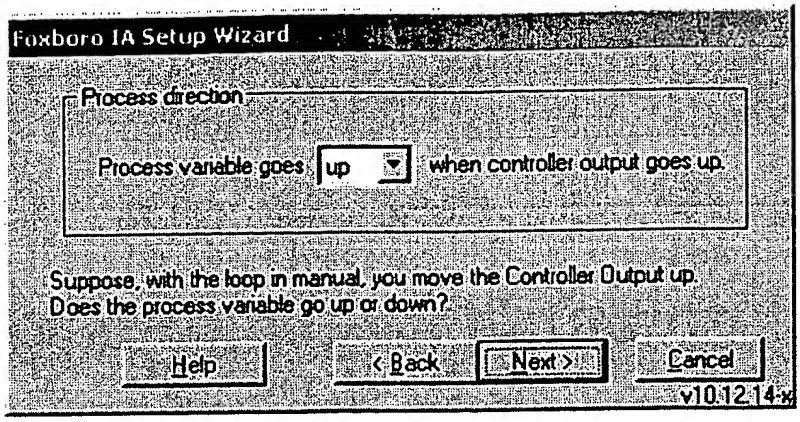


Figure 38 is where the Foxboro IA setup wizard which has been created by the wizard builder invention is asking for the Process Direction. When the user was using the wizard builder invention to create the Foxboro IA setup wizard he indicated that the Process Direction could not be read automatically from the loop controller and must be entered manually. See figure 30. Therefore when the Foxboro IA setup wizard is running, it asks the user for the process direction.

Figure 39. The created wizard asks for the engineering units

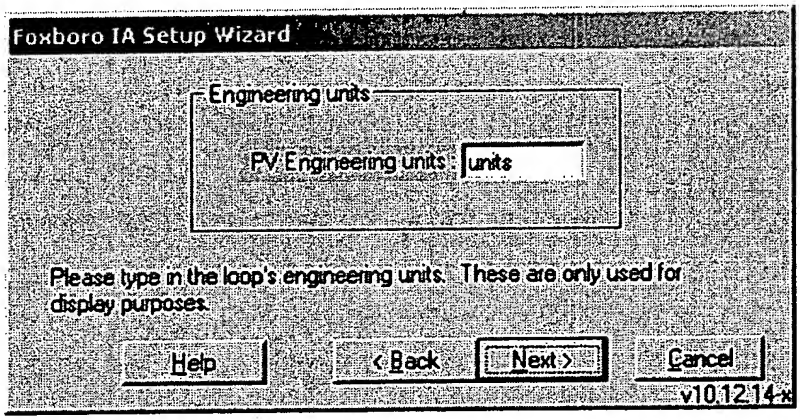


Figure 39 is where the Foxboro IA setup wizard, which has been created by the wizard builder invention, is asking for the loop's engineering units. In this early implementation of the wizard builder invention, the created setup wizard always asks for the engineering units. In later implementations, the user of the wizard builder invention has the option of setting the engineering units to be read automatically from the loop controller.

Figure 40. The created wizard asks for the sample interval

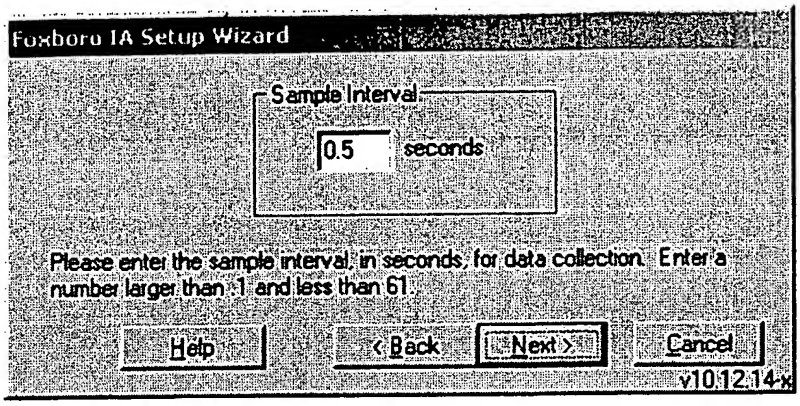


Figure 40 is where the Foxboro IA setup wizard, which has been created by the wizard builder invention, is asking for the loop's sample interval. The created setup wizard always needs to ask for the loop's sample interval. It can never be read automatically from the loop controller.

Figure 41. The created wizard's summary screen

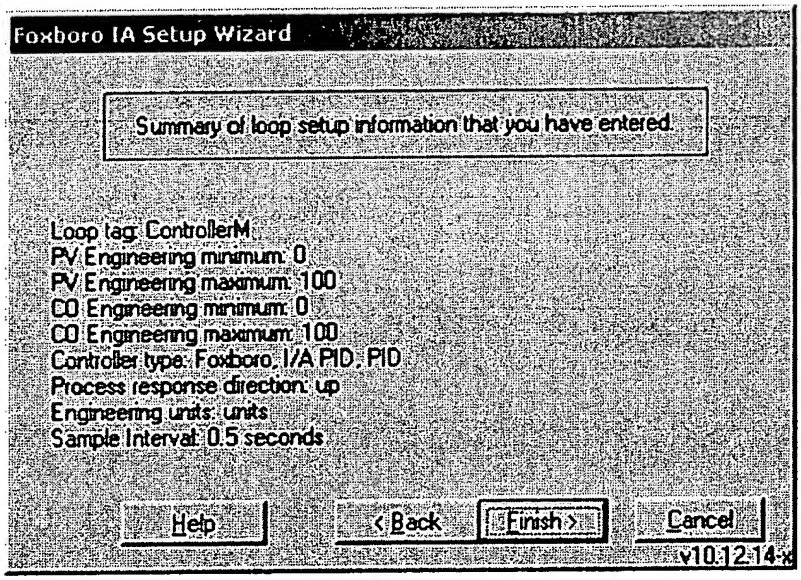


Figure 41 is the summary screen for the Foxboro IA setup wizard, which has been created by the wizard builder invention. When the user clicks finish, he will have finished setting up a de-facto DDE interface to his loop controller.

Figures 1 to 3 prove that the wizard builder invention, demonstrated in figures 4 to 41, existed prior to 24 November 2000. The screenshots in figures 4 to 41 have been generated by compiling and running source code that existed 24 November 2000 and have been maintained by the Microsoft SourceSafe system since that date. Figures 4 to 32 are screenshots from the wizard builder invention creating a new setup wizard named "Foxboro IA". This was done by asking questions in plain English. The user does not need knowledge of writing or using computer programs to answer the questions. Figures 33 to 41 are screenshots obtained when the created setup wizard itself runs.